

Bankiet

Dostępna pamięć: 32 MB.

W restauracji Utalentowany Miś zaplanowano bankiet dla finalistów OIG. Goście zasiadają przy okrągłych stołach w ściśle określony sposób. Kierownik sali otrzymał listę gości wraz z informacją, kto ma siedzieć z lewej strony każdego z nich. Ile stołów musi przygotować na bankiet?

Zadanie

Opracuj program, który:

- wczyta ze standardowego wejścia informacje o rozmieszczeniu gości,
- obliczy ile stołów trzeba przygotować,
- wypisze na standardowe wyjście wynik.

Wejście

W pierwszym wierszu zapisano liczbę gości N ($1 \leq N \leq 30.000$). Goście są ponumerowani kolejnymi liczbami naturalnymi od 1 do N .

W drugim wierszu zapisano numer gościa siedzącego po lewej stronie pierwszego gościa. w trzecim wierszu zapisano numer gościa siedzącego po lewej stronie drugiego gościa itd.

W i -tym wierszu zapisano numer gościa siedzącego po lewej stronie $(i - 1)$ -tego gościa.

W $N + 1$ -szym wierszu zapisano numer gościa siedzącego po lewej stronie N -tego gościa.

Wyjście

W pierwszym wierszu wypisz liczbę stolików potrzebnych do usadzenia wszystkich gości.

Przykład

Dla danych wejściowych:

```
12
4
10
7
3
2
6
1
5
11
8
12
9
```

poprawnym wynikiem jest:

4

Rozwiązanie

Mając podaną osobę siedzącą na lewo od każdego z gości możemy obsadę każdego ze stołów przedstawić w postaci cyklu. Dla przykładu weźmy gościa pierwszego z testu przykładowego. Po jego lewej stronie musi znaleźć się gość czwarty, dalej gość trzeci itd. Pełny cykl wygląda następująco:

1 -> 4 -> 3 -> 7 -> 1

Ponieważ każdy gość siedzi przy którymś stole, to każdy należy do dokładnie jednego cyklu. Zauważmy, że niektóre cykle mogą być jednoelementowe: w teście przykładowym na lewo od gościa #6 siedzi gość #6, czyli tak naprawdę gość szósty sam przesiaduje samotnie.

Jak łatwo zauważyć, ilość stołów potrzebnych do obsługi bankietu równa jest ilości takich cykli. Równie łatwo przebiega konstrukcja algorytmu zliczającego zliczającego cykle:

1. oznacz wszystkich gości jako nieprzetworzonych
2. iteruj po wszystkich gościach; jeśli dany gość jest nieprzetworzony, idź do punktu 3.
3. oznacz danego gościa oraz wszystkich należących do tego samego cyklu jako przetworzonych
4. zwiększ ilość stołów o jeden i wróć do punktu 2.

Ponieważ każdego gościa badamy/oznaczamy jako przetworzonego tylko raz, uzyskujemy algorytm działający w czasie liniowym względem ilości gości. Dla każdego gościa zapamiętujemy jego następnika przy stole oraz stan w algorytmie (przetworzony lub nie), więc złożoność pamięciowa także wynosi $O(n)$.